In The Claims

Please amend the claims as follows:

1. (Amended) A computing device executing a graphics rendering software program ~~for~~ providing instructions to one or more processors to render graphics on a display ~~of an embedded~~ , the computing device configured ~~for establishing~~ to establish a network connection with at least one other computing device, comprising:

~~(a) an application layer;~~

~~(b) a graphics toolkit; and~~

~~(c)~~ a graphics driver ~~for~~ rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on ~~a~~ the display ~~of the embedded computing device~~, and

~~(d)~~ wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

~~(e)~~ wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of rectangular clip segments, and

wherein the graphics driver is further configured to increment a visibility tag corresponding to the second drawing surface when a z-order of the second drawing surface is changed, and to compute a new set of rectangular clip segments if the visibility

tag corresponding to the second drawing surface is not the same as a visibility tag corresponding to the first drawing surface.

2. (Canceled).

3. (Canceled).

4. (Amended) The computing device ~~graphics rendering software program~~ of claim 1, wherein the first drawing surface and the second drawing surface, including both the visible portion and an obscured portion, comprise rectangular borders.

5. (Amended) The computing device ~~graphics rendering software program~~ of claim 1, wherein each rectangular clip segment of the set of rectangular clip segments is iteratively output to the display for displaying the visible portion of the second drawing surface.

6. (Amended) A computing device executing a graphics rendering software program ~~for~~ providing instructions to one or more processors to render graphics on a display ~~of an embedded~~ ,the computing device configured ~~for establishing~~ to establish a network connection with at least one other computing device, comprising:

~~(a) an application layer~~;

~~(b) a graphics toolkit; and~~

~~(c)~~ a graphics driver ~~for~~ rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on ~~a~~ the display ~~of the embedded computing device~~, and

~~(d)~~ wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

~~(e)~~ wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of rectangular clip segments, and

A1 cont

~~(f)~~ wherein the set of rectangular clip segments is stored as a graphics context object corresponding to unobscured segments of the second drawing surface, and

wherein the graphics driver is further configured to increment a visibility tag corresponding to the second drawing surface when a z-order of the second drawing surface is changed, and to compute a new set of rectangular clip segments if the visibility tag corresponding to the second drawing surface is not the same as a visibility tag corresponding to the first drawing surface.

7. (Canceled).

8. (Canceled).

9. (Amended) The computing device ~~graphics rendering software program~~ of claim 6, wherein the first drawing surface and the second drawing surface, including both the visible portion and an obscured portion, comprise rectangular borders.

10. (Amended) A computing device executing a graphics rendering software program ~~for~~ providing instructions to one or more processors to render graphics on a display ~~of an embedded~~ ,the computing device configured ~~for establishing~~ to establish a network connection with at least one other computing device, comprising:

A1 Cont

~~(a) an application layer~~;

~~(b) a graphics toolkit; and~~

~~(c)~~ a graphics driver ~~for~~ rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on ~~a~~ the display ~~of the embedded computing device~~, and

~~(d)~~ wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

~~(e)~~ wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of rectangular clip segments, and

(f) wherein the set of rectangular clip segments is stored as a graphics context object corresponding to unobscured segments of the second drawing surface, and

(g) wherein each rectangular clip segment of the set of rectangular clip segments is iteratively output to the display for displaying the visible portion of the second drawing surface, and

wherein the graphics driver is further configured to increment a visibility tag corresponding to the second drawing surface when a z-order of the second drawing surface is changed, and to compute a new set of rectangular clip segments if the visibility tag corresponding to the second drawing surface is not the same as a visibility tag corresponding to the first drawing surface.

A1 cnt

11. (Canceled).

12. (Canceled).

13. (Amended) The computing device ~~graphics rendering software program~~ of claim 10, wherein the first drawing surface and the second drawing surface, including both the visible portion and an obscured portion, comprise rectangular borders.

14. (Amended) A computing device executing a graphics rendering software program ~~for~~ providing instructions to one or more processors to render graphics on a display ~~of~~

~~an embedded~~ ,the computing device configured ~~for establishing~~ to establish a network connection with at least one other computing device, comprising:

~~(a) an application layer;~~

~~(b) a graphics toolkit; and~~

~~(c)~~ a graphics driver ~~for~~ rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on ~~a~~ the display ~~of the embedded computing device~~, and

~~(d)~~ wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

A1 cmt

~~(e)~~ wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of clip segments, and

wherein the graphics driver is further configured to increment a visibility tag corresponding to the second drawing surface when a z-order of the second drawing surface is changed, and to compute a new set of rectangular clip segments if the visibility tag corresponding to the second drawing surface is not the same as a visibility tag corresponding to the first drawing surface.

15. (Amended) The computing device ~~graphics rendering software program~~ of claim 14, wherein the graphics driver includes:

(1) a shape function layer including a target architecture specific instruction set ~~for~~ setting and retrieving pixel ~~values~~ numbers, respectively, into and from a one-dimensional framebuffer memory; and

(2) a framebuffer access macro layer including a set of macros ~~for~~ inlining into the shape function layer.

16. (Amended) A computing device executing a graphics rendering software program ~~for~~ providing instructions to one or more processors to render graphics on a display of an embedded computing device configured ~~for establishing~~ to establish a network connection with at least one other computing device, comprising:

A1 Cnt

~~(a) an application layer;~~

~~(b) a graphics toolkit; and~~

~~(c)~~ a graphics driver ~~for~~ rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on ~~a~~ the display ~~of the embedded computing device~~, and

~~(d)~~ wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

~~(e)~~ wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of clip segments, and

~~(f)~~ wherein the set of clip segments is stored as a graphics context object corresponding to unobscured segments of the second drawing surface, and

wherein the graphics driver is further configured to increment a visibility tag corresponding to the second drawing surface when a z-order of the second drawing surface is changed, and to compute a new set of rectangular clip segments if the visibility tag corresponding to the second drawing surface is not the same as a visibility tag corresponding to the first drawing surface.

17. (Amended) The computing device ~~graphics rendering software program~~ of claim 16, wherein the graphics driver includes:

A1 Cont

(1) a shape function layer including a target architecture specific instruction set ~~for~~ setting and retrieving pixel ~~values~~ numbers, respectively, into and from a one-dimensional framebuffer memory; and

(2) a framebuffer access macro layer including a set of macros ~~for~~ inlining into the shape function layer.

18. (Amended) A computing device executing a graphics rendering software program ~~for~~ providing instructions to one or more processors to render graphics on a display ~~of an embedded~~ ,the computing device configured ~~for establishing~~ to establish a network connection with at least one other computing device, comprising:

~~(a) an application layer;~~

~~(b) a graphics toolkit; and~~

~~(c)~~ a graphics driver ~~for~~ rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on ~~a~~ the display ~~of the embedded computing device~~, and

~~(d)~~ wherein the graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display, and

~~(e)~~ wherein, when the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of clip segments, and

~~(f)~~ wherein the set of clip segments is stored as a graphics context object corresponding to unobscured segments of the second drawing surface, and

~~(g)~~ wherein each clip segment of the set of clip segments is iteratively output to the display for displaying the visible portion of the second drawing surface.

wherein the graphics driver is further configured to increment a visibility tag corresponding to the second drawing surface when a z-order of the second drawing surface is changed, and to compute a new set of rectangular clip segments if the visibility tag corresponding to the second drawing surface is not the same as a visibility tag corresponding to the first drawing surface.

19. (Amended) The computing device ~~graphics rendering software program~~ of claim 18, wherein the graphics driver includes:

(1) a shape function layer including a target architecture specific instruction set ~~for~~ setting and retrieving pixel ~~values~~ numbers, respectively, into and from a one-dimensional framebuffer memory; and

(2) a framebuffer access macro layer including a set of macros ~~for~~ inlining into the shape function layer.

20. (Amended) A method of rendering graphics including overlapping drawing surfaces on a display of an embedded computing device configured for establishing a network connection with at least one other computing device, comprising the steps of:

(a) computing a set of clip segments corresponding to a visible portion of a partially obscured drawing surface; and

(b) rendering the partially obscured drawing surface along with an overlapping drawing surface on the display, and

(c) incrementing a visibility tag corresponding to the partially obscured drawing surface when a z-order of the partially obscured drawing surface is changed, and computing a new set of rectangular clip segments if the visibility tag corresponding to the partially obscured drawing surface is not the same as a visibility tag corresponding to the overlapping drawing surface.

21. (Original) The method of claim 20, wherein the clip segments correspond to rectangular portions of the visible portion of the partially obscured drawing surface.

22. (Original) The method of claim 21, wherein the partially obscured drawing surface and the overlapping drawing surface comprise rectangular borders.

23. (Original) The method of claim 21, further comprising the step of storing the set of clip segments as a graphics context object corresponding to unobscured segments of the partially obscured drawing surface.

A1 cont

24. (Original) The method of claim 23, wherein the clip segments correspond to rectangular portions of the visible portion of the partially obscured drawing surface.

25. (Original) The method of claim 24, wherein the partially obscured drawing surface and the overlapping drawing surface comprise rectangular borders.

26. (Original) The method of claim 23, further comprising the step of iteratively outputting each clip segment of the set of clip segments to the display for displaying the visible portion of the partially obscured drawing surface.

27. (Original) The method of claim 26, wherein the clip segments correspond to rectangular portions of the visible portion of the partially obscured drawing surface.

28. (Original) The method of claim 27, wherein the partially obscured drawing surface and the overlapping drawing surface comprise rectangular borders.